
Appendix B: MTA diagnostics reference

This appendix consists of a command reference for the Modular Test Architecture (MTA) diagnostics. Tests described in this manual do not directly apply to a specific installation.

MTA Testview

Testview is the name of the menu interface for MTA test programs and support files.

Two interfaces exist for the MTA tests. T (t.exe) is the command line interface for the tests, while Testview is a menu interface. Both interfaces allow users to run test sequences through a numerical mapping. In both cases a PKG file is used to map names and numbers to a test invocation line.

Testview main menus

When you sign on to Testview, a menu bar appears across the center of the screen. **Table B-1** lists the available menus.

Table B-1
Testview menus

Menu name	Function
TEST	Access to test modules and subtests.
ERRORS	View history of error messages.
SUMMARY	View summary of test run.
CONFIGURATION	Menu and system configuration tools.
OPTIONS	Changes the way tests are run.
HELP	Online help for using Testview menus.
QUIT	Exits Testview and returns to DOS.

README.TXT README.TXT is a file distributed with each release of the diagnostic package. You may view this file by pressing <Enter>, not F1, from the Testview menus.

Using T.EXE T (t.exe) is used to run tests from the DOS command line or the batch file. For more information see t.hlp or press <Enter>, not F1, on the appropriate Testview menu item.

Starting tests

There are several ways to initiate diagnostic tests in Testview. [Table B-2](#) summarizes the possibilities. If you press F2 or ALT-F2, you are prompted to specify the number of loops. If you press <Enter> while in a SUBTEST menu, the test will run once only.

Table B-2
Test module options

Test Parameters	TEST menu	SUBTEST menu
All modules	ALT-F2	---
One module	F2	ALT-F2
One test	---	F2 or ENTER

Execution sequence

The tests will execute in the order in which they appear in the menus. This order may be changed using Ctl+D and Ctl+A or by editing the PKG file. The subset of tests may be changed by using F3 and F4 to enable or disable subtests or modules.

The *Run-list* selection in the **OPTIONS** menu allows you to specify different sequences of tests with looping. **Table B-3** summarizes how to set parameters in the run-list buffer menu.

Table B-3
Run-list functions

Run list setting	Function
1-8	Range of modules
1.3-1.5	Range of subtests
1*5	Use looping constant
1-4*5	Module range with looping constant
12*	No looping constant
{1,2,3}	Sequence of modules
{1,2,5.1,3*3,4}	Sequence of modules and subtest with looping
{LED,CPU,DRAM}	Names may be used

Note: Do not put spaces between commas and module names and numbers.

Looping options

There are three ways to run multiple sequences of tests:

- The F2 function key prompts you for the number of loops before executing.
- You can also change the invocation parameter /N. Changing /N will cause the test to run multiple times each time its name comes up in a sequence.
- The run-list may also be used to loop arbitrary test sequences.

Test window

The output of each test appears in a window. Certain test results, such as the video test, use a full-screen window. If the result output exceeds the window area, you can use F7 or the **ERROR** menu to review the results.

When a test is complete, the “press any key to continue” prompt displays.

Error messages

Error messages are directed to the console or to a test window. You can redirect the error message output by changing the /Rx run-time flag (where x can be R, E, A, or S) through the OPTIONS menu or the F5 function key.

Messages redirected to a file can be viewed through the ERRORS menu or function key F7. The default setting of the error message buffer file is set to record error messages only. If “status” or “advisory” messages are desired, use F5 to change the global run-time flag from “/RE test.out” to “/RR test.out”.

Runtime enviroment

When booted from diskette, **Testview** runs from the RAM drive. The AUTOEXEC.BAT file on the boot diskette sets up a RAM drive before invoking **Testview**. **Testview** uses the RAM drive as a virtual cache by copying tests selected for execution to the RAM drive, and then running the tests. Tests will execute faster if they are running from RAM.

Once a test EXE file is loaded, it remains on the RAM drive. If space on the RAM drive becomes full, Testview removes tests from the RAM drive before copying a new test to the RAM drive.

Testview and all of its test module files can be copied to a hard drive or another floppy and executed from there. It will run exclusively out of the directory in which it was placed.

Using the RAM drive is faster, but any files which you have modified will be purged when the Application Processor is shut down. You can avoid this situation by specifying a pathname on nonvolatile media when possible, or by using DOS to copy files.

Stopping tests

Using Ctrl-C or Ctrl-Break will stop tests. The Ctrl-C command will not work if the input buffer is full. The Control-Break command is more effective, but may interrupt critical DOS functions.

The amount of time that a test takes to respond to Ctrl-C depends on the type of loop that is running and the amount of clean-up required.

Run-time flags

The execution of diagnostic tests is controlled through run-time flag settings. **Table B-4** summarizes the flag types.

Table B-4
Run-time flags in Testview

Flag type	Flag name
required	/T /CF
control	/HE /PE /N /RR /RE /RA /RS /Q /SS /SF /W
custom	/P /V /L /CD and others

Table B-5 summarizes the required run-time flag types.

Table B-5
Required run-time flags

Flag name	Function
/T	Required for all MTA tests.Specifies the internal subtest number for the test module.
/CF	Specifies the CFG file. Not necessary if a default CFG file is present. The default file name is the base name of the EXE file with a CFG suffix.

The control flags are handled by the MTA run-time library. [Table B-6](#) summarizes the available run-time control flags.

Table B-6
Run-time control flags

Flag name	Function
/HE	Halt on error. Halts testing when an error occurs.
/PE	Prompt on error. If an error occurs, the test will not proceed until keyboard input is received.
/N	Specifies number of times to run the test. Defaults to “/N 1”, not needed with Testview.
/Q	No messages to console. Does not affect the other redirects.
/RR file	Redirect error, advisory, and status messages to file.
/RE file	Redirect error messages to file.
/RA file	Redirect advisory messages to file.
/RS file	Redirect status messages to file.
/SS <file>	Summarize loops and fails after each test. Can be optionally redirected to a file or to a printer.
/SF file	Summary file redirection. Overrides default.
/W nn	Wait only nn seconds when reading keyboard. Waits indefinitely if the parameter is not present. NN may be any positive integer value. This is useful for running interactive tests unattended.

Custom flags are handled directly by the tests and may not behave the same way for each test. For information on custom flags for a specific test, refer to the On-Line-Help file for the test module.

Sources of Run-time flags

The parameters that control testing can be found in different sources. They are listed below in priority order. If there is a conflict between flags, the flag specified from a higher priority is selected.

- the global run-time flags buffer in the menus
- Testview or T command line
- the subtest entry in the PKG file
- the module entry in the PKG file

Flags entered on the Testview command line are placed in the global run-time flags buffer along with the Testview default (/RE test.out). This buffer is modified with the F5 function key. The buffer is combined with the module and subtest entries from the PKG file before processing. The module and subtest run-time flag entries are modified with the F5 function key.

CFG parameters

Each MTA test is designed to handle device and peripheral implementations using hardware configuration parameters instead of hard coding. The specific parameters vary among the test modules, but each is described by its label and assigned a value.

The parameters are found in the PKG file or in the CFG file. The *ts_getcfg_xxx* calls access the parameters in both locations. Parameters in the PKG file take precedence over those in the CFG file.

The default CFG file has the same base name as the load file. It has a .cfg extension instead of .exe. If the file name is different from the load file, use the flag */CF <file name>* on the test invocation line.

The file contains one line for each parameter and comments. The format of a line specifying a parameter is

```
<module name>.<parameter name> = <value>
```

The value can be a number or a string. Quotation marks will force a number to be interpreted as a string or will force spaces to be imbedded in the string. The maximum length of a value string is 256 characters.

Pressing the F6 function key displays an editable parameter list. The listing includes the module name in square brackets followed by the parameter. The CFG parameters are listed before PKG parameters.

If the changes are saved, a file of the same name is written to the RAM drive. Changes will apply to the current session only. To save changes to the CFG file, exit to DOS and copy the changed file to the floppy.

PKG file

The PKG file is an ASCII (readable) file containing a description of each module and subtest in the package. The PKG file is separated into groups of subtests belonging to modules. In addition to subtests, module groups may contain CFG parameters. The file may also contain one or more product-code sections.

The PKG file creates the logical mapping between test names and their corresponding EXE files and parameters. The order of the subtests and modules in the file determines the numerical labeling. Use the numerical labels to reference tests when using the T.EXE command.

PKG is not case-sensitive and may contain comments. Comments appear on separate lines between the “/*” and “*/” symbols.

Subtest syntax

There are two ways of specifying a subtest. The preferred method is based on the Windows **.ini** format, where each group of subtests is preceded by a header. For example:

```
[disk_drive_1, drive.exe /cf mongo.cfg]
test1, +w, /t 1
test2, +w, /t 7
test3, +wid, /t 2
```

The header contains the module name and an optional field with common invocation flags. The subtest line consists of three fields: the subtest name, attribute flags, and subtest-specific invocation flags. Invocation flags are combined with the module and the subtest.

You can also specify a subtest by placing all information on a single line, as in the following example:

```
disk_drive_1.test1, +w, drive.exe /cf mongo.cfg /t 1
disk_drive_1.test2, +w, drive.exe /cf mongo.cfg /t 7
disk_drive_1.test3, +wid, drive.exe /cf mongo.cfg /t 2
```

Cloning

Some modules represent hardware components, such as hard drives, which have more than one unit present in the system. The following syntax directs Testview to create a second module identical to the first.

```
[disk_drive_2<disk_drive_1>]
```

Attribute flag

The attribute flags, which consist of one or more single characters, cause a warning to occur when a test is enabled by an operator. The following table lists the attribute flags available.

Table B-7
Attribute flags

Flag	Function
+	subtest enabled
-	subtest disabled
C	user configuration required
D	destructive subtest
H	external hardware required
I	interactive subtest
W	run in a window (default is full screen)

The attribute flag must always start with either + or -, but other characters may occur in any order. The **W** flag is expected for all tests except those which interfere with the display, such as video tests.

Product_code sections

Product_code sections contain CFG parameters that override the defaults and configure the system to a specific product code. For example:

```
.  
. /* normal stuff here */  
.  
  
[[product_code_X]]  
module.label = value  
module.mable = value  
module.nable = value  
  
[[product_code_Y]]  
.  
.  
.
```

The */PC <product code>* command line flag to t or Testview will cause the statements in the referenced section to be evaluated. Otherwise, the sections are ignored.

Minus (-) Module or Subtest

Most of the product code sections specify system configurations with fewer assembly subunits than the total possible. Use the minus syntax to exclude subunits not expected in a specific product code.

-MODULE

-MODULE.SUBTEST

This syntax excludes the module or its subtests from the menus and the run list.

The product code section can enable a subtest which is normally disabled using the plus (+) sign.

+MODULE.SUBTEST

The + syntax does not have a corresponding effect on modules.

MODULE NAME

The name of the module, up to a maximum of 31 characters.

SUBTEST NAME

The name of the subtest, up to a maximum of 31 characters.

Attributes

The characteristics field is a string of characters from the subtest entry in the PKG file. The subtest entry in a PKG file has the form

```
<logical name>, <characteristics>, <invocation line>
```

Table B-8 summarizes the defined single-character flags.

Table B-8
Single-character attributes

Flag	Function
+	subtest enabled
-	subtest disable
W	run in a window (default is full screen)
D	destructive subtest
I	interactive subtest
H	external hardware required to run subtest
C	configuration files required to run subtest

The flag string must always start with either + or -, but other flags may occur in any order. The W flag is expected for all tests except those which write directly to the console. Not using W, reduces the risk of file corruption.

The attribute flags display a warning whenever the operator attempts to enable the subtest. Any number of these flags may occur in the string.

Invocation

The invocation line is a field from the subtest entry in the current PKG file. A subtest entry in the PKG file has the form

```
<logical name>, <attributes>, <invocation line>
```

A module entry has the form

```
[<module name>, <invocation line>]
```

The invocation line is a string which is passed to DOS when the test is invoked. It starts with the **.exe** file name, followed by a list of flags.

The /T flag is required and is used by test modules to determine which subtest to run.

```
"floppy.reset, +W, flop.exe /T 3"
```

or

```
"[floppy, flop.exe]  
reset, +W, /T 3"
```

The two invocation lines are concatenated, module first, subtest second. The flags are defined under **"Using the T Command"** on page -29.

Function keys

The function keys are available while you are in the TEST and SUBTEST menus. However, the F1 function key produces help for selected items.

Table B-9
Function keys

Key name	Function
F1 and ALT-F1	Help for <i><item></i>
F2 and ALT-F2	Run <i><item></i>
F3 and ALT-F3	Disable <i><item></i>
F4 and ALT-F4	Enable <i><item></i>
F5	Edit run-time flags for <i><item></i>
F6 and ALT-F6	Edit configuration parameters for <i><item></i>
F7	View error messages
F8	View summary
F9 and ALT-F9	Display version of <i><item></i>
F10	Print <i><help file or parameter list></i>

Using menus

There are three menu types used in **Testview**:

- selectable lists
- text windows
- value lists

These menus are described in the following sections.

Selectable Lists

These menus contain a list of items.

The menu may contain more items than can fit in the window. An arrow icon appears on the window border to indicate the undisplayed items.

Table B-10 lists the keys that can be used to change the active position in the menu.

Table B-10
Testview selection keys

Key	Function
ENTER	Differs for each list
UP ARROW	Moves up by one (cycles from 1 to last)
DOWN ARROW	Moves down by one (cycles from last to 1)
PAGE UP	Moves up by size of window
PAGE DOWN	Moves down by size of window
HOME	Moves to first item
END	Moves to last item
"1,2..a,b"	Jumps to next item starting with character
ESCAPE	Closes menu and returns to previous menu.
F1	Displays help information about the item.

Text Windows

Text windows display text for help, summary, and error messages.

Table B-11 lists the keys used to reposition text in the window.

Table B-11
Text positioning keys

Key	Function
UP ARROW	Moves position up by one
DOWN ARROW	Moves position down by one
PAGE UP	Moves position up by size of window
PAGE DOWN	Moves position down by size of window
SPACE BAR	Moves position down by size of window
HOME	Moves position to top of file
END	Moves position to end of file
ESCAPE	Closes window, returns to menu

Value lists

Data entry forms are used to modify parameters, such as the hardware configuration.

Table B-12
Data selection keys

Key	Function
ENTER	Save your changes. A dialog box prompts you for verification.
ESCAPE	Exit without saving changes.
INS	Toggles between insert and overwrite modes.
HOME	Jumps to the first character in the line.
END	Jumps to the last character in the line.
ARROWS	Move the cursor position around the form.

Test module menu

The TEST menu lists the available test modules. The list of modules reflects the current system configuration. The maximum number of modules is 99.

Table B-13 describes the keys available in this menu.

Table B-13
Test module keys

Key	Function
ENTER	Opens the SUBTEST menu. A SUBTEST menu is a list of all the subtests available in this module.
SPACE	Toggles enable/disable settings for all the tests in the module. That is, if all subtests are disabled, pressing SPACE is the same as pressing F4. If any of the subtests are enabled, pressing SPACE will disable all of them.
F1	Displays the entire help file. The help file contains general information about the test module and specific information about each subtest.

Table B-13
Test module keys (Continued)

Key	Function
F2	Runs all enabled tests in the module. A dialog box prompts for the number of loops to execute (the number 0 means no limit).
F3	Disables all tests in the selected module. Disabled means it cannot be run until it is reenabled.
F4	Enables all tests in the selected module. If any of the tests have secondary attributes (see the section below on attributes), confirmation will be requested. If a test has more than one attribute, more than one confirmation will be requested. Use <Ctrl+C> to stop the enable sequence.
F5	Accesses the global run-time flags and module run-time flags. (To change the subtest run-time flags use F5 in the SUBTEST menu.)
F6	Accesses the test module CFG parameters. The CFG file is displayed (if one is specified) and any parameters from the PKG file. The parameters from the PKG file will have brackets ([]) around the module name.
F7	Displays current error message file.
F8	Displays current summary file.
F9	Displays the version number of the test module.
ALT+F2	Runs all enabled tests in all modules. As in F2, a dialog box prompts for the number of loops to execute.
ALT+F3	Disables all tests in all modules.
ALT+F4	Enables all tests in all modules. The handling of secondary attributes, such as “destructive” by the ALT+F4 key, is the same as F4 above.
ALT+F6	Accesses all CFG parameters read from the PKG file. This includes system parameters, such as “bios_id” or “version_number”, as well as module parameters.

Table B-13
Test module keys (Continued)

Key	Function
ALT+F9	Displays the current version number of Testview .
CTL+A	Adds a new item to the TEST menu from the delete-buffer.
CTL+D	Deletes an item from the TEST menu, copies it to the delete-buffer
CTL+E	Edits the module name and module invocation parameters.
End	

Subtest menu

The SUBTEST menu contains the list of subtests available in the selected module. The specified number of subtests should be between 1 and 99. You can scroll through the list and select any entry using the arrow and page keys.

Table B-14
Subtest menu keys

Key	Function
ENTER	Runs the selected test once. If the test is disabled, a warning message will appear. Pressing <Cntrl+C> aborts the test. (see the "Stopping tests" section).
ESCAPE	Causes the SUBTEST menu to disappear, and the active position returns to the TEST menu.
SPACE	Toggles enable and disable for the selected subtest.
F1	Displays the portion of the test module help file which applies to the selected subtest.
F2	Runs the subtest. A dialog box prompts for the number of loops to execute (the number 0 means no limit).
F3	Disables the selected subtest.

Table B-14
Subtest menu keys (Continued)

Key	Function
F4	Enables the selected subtest. (See the description of how attributes affect F4 in the “PKG file” section).
F5	Accesses the run-time flags from all sources.
F6	Accesses the test module CFG parameters. The CFG file is displayed (if one is specified) and any parameters from the PKG file. The parameters from the PKG file will have brackets ([]) around the module name.
F7	Displays error message file.
F8	Displays summary file.
F9	Displays the version number of the test module.
ALT+F2	Runs all subtests in the module. Like F2, a dialog box allows the user to specify the number of loops to run the number of loops prior to execution.
ALT+F3	Disables all subtests in the menu.
ALT+F4	Enables all subtests in the menu. <i>Note:</i> This may result in many attribute warnings. To stop the prompting sequence, press <Ctrl+C>.
ALT+F6	Accesses all CFG parameters read from the PKG file. This includes system parameters, such as “bios_id” or “version_number”, as well as module parameters.
ALT+F9	Displays the current version number of Testview .
CTL+A	Adds a new subtest to the menu from the delete-buffer in reverse order (see CTL+D).
CTL+D	Deletes subtest, inserts it in delete-buffer.
CTL+E	Edits the subtest name, attribute flags, and run-time flags from the PKG file. See the section on “PKG files” for more information.
End	

Errors menu

Use the Errors menu to review error messages. The menu has four options: View, Clear, Save, and Redirect. The View option is also available through the F7 function key.

This menu uses the redirect parameter on the primary invocation parameter list. If the parameter is changed (using the F5 function key) the menu will use the new file name. If the parameter is removed or if output is redirected to the line printer, the menu is disabled. If there is more than one redirect file specified, the first one is used.

The default destination of the errors is a file named “test.out” located on the RAM drive. This file disappears when the system is rebooted and must be saved manually in DOS if desired.

Table B-15 summarizes the Error menu options.

Table B-15
Errors menu options

Option	Function
View Errors	Displays the current error message file.
Clear Errors	Deletes any text in the file and resets the flashing ERRORS box on the display.
Save Errors	Copies the errors file to a new location, or transfers the file from the RAM disk to the floppy.
Redirect Errors	Changes the name and the location of the error file.

Summary menu

The summary file contains a list of completed tests, test passes, test failures, and error messages. The total number of times run is the sum of pass count and fail count. The number of errors may on some occasions exceed the number of failures.

The summary file is named “test.sum” and is located on the RAM drive. If Testview is running from the RAM drive, the file disappears when the system is rebooted.

Table B-16 details the Summary menu options.

Table B-16
Summary menu options

Option	Function
View Summary	Displays the current summary file.
Clear Summary	Deletes any text in the file.
Save Summary	Copies the Summary file to a new location, or transfers the file from the RAM disk to the floppy.
Redirect Summary	Changes the name and location of the summary file.

Configuration menu

You can customize your menus with different test packages, or different colors, or change the *testview.ini* file through the Configuration menu. The contents of the Test and Subtest menus are determined by the PKG file.

Configure test menu from list

The *Configure test menu from list* selection offers a menu of different configuration choices. Each entry represents a file on the boot device in **pkg** format. The actual words in the menu are read from a line in the **pkg** file. If a selection is made, the current test menus are replaced with the contents of the selected file.

Select a Configuration Submenu

This menu contains a list of descriptions of all files with a **.pkg** suffix.

Pressing <Enter> after selecting one of the items invokes the Test menu. If the file contains product codes, another submenu appears allowing you to select a product code.

Pressing <F1> on a selected item displays its file name.

Select a Product Code Submenu

This menu appears if the selected configuration file contains product codes. A product code contains parameters which describe hardware in a specific system. If a product code is unknown, press <Esc> to exit the menu.

Configure test menu from file

This choice invokes a dialog box requesting a test menu file name. The system replaces the current test menus with the contents of the requested file.

Self-configure test menu

This choice configures the test menus based on the hardware detected in the system. The BIOS-id for the current system is detected and a pkg file is selected based on the ID. The self-sense test for each module is run and may alter some of the CFG parameters.

Auto-integrate test package

This option runs the same utilities found in the Self-configure test menu, as well as the integration utilities configured in the Integration_List parameter in the Package module. After running the utilities, it prompts for a product code and writes the altered cfg parameters and the generated run-list to the pkg file.

Display system hardware

This option probes the system hardware and builds the display. If the file *display.cfg* exists, its contents will be displayed without probing.

Save test configuration

The *Save test configuration* option converts current test menus to pkg format and writes the data to a file. You are prompted to enter a file name and title.

Change menu colors

This option provides access to menu colors through lists of changeable menu regions. You can save or restore your selections.

The following display options are available:

- Desk top
- Main menus
- Form window
- Text window
- Test window
- Dialog box

Selecting one of the above options invokes a menu listing of specific display regions. The following display regions are available:

- Background
- Letters
- Enabled letters
- Disabled letters
- Border background
- Border lines

There are eight available colors in VGA text mode including black and white. Each of these colors has a blink or bold option depending on whether it is used in the foreground or background.

Table B-17 lists the keys used in Change menu colors.

Table B-17
Change menu colors options

Key	Function
SPACE BAR	Steps through each of the color choices
+ (plus sign)	Toggles bold attribute for letters and lines only
- (minus sign)	Toggles blink attribute for letters and lines only
ENTER	Saves colors for the current Testview session
ESCAPE	Exits menu without saving changes

Color changes remain in effect only for the current session unless the Save colors option is used.

The Save colors and Restore colors options access the color settings from the file **TESTVIEW.INI**.

Change TESTVIEW.INI parameters

This choice displays a menu of the parameters in the TESTVIEW.INI file.

Options menu

The Options menu presents a list of miscellaneous choices. Some are available through function keys, some apply to the tests, and others apply to the menus only.

Line printer

This option toggles error message redirection between the line printer and the default *test.out* file.

Halt on error

This option toggles the halt on error flag (“/HE”).

Prompt on error

This option toggles the prompt on error flag (“/PE”).

Summarize Loops/Fails

This option toggles the set summarize flag (“/SS”).

Interactive waiting

This option allows you to set a response waiting period for an interactive test. Most modules contain one or more tests requiring operator input to pass. If there is no operator, the tests will wait indefinitely. When this option is selected, the test will wait for the number of seconds specified.

Edit run-time flags

This option provides access to global run-time flags. The buffer contains any flags present on the Testview command line as well as the defaults added by /RE test.out. **Table B-18** lists the run-time flags supported by the run-time library.

Table B-18
Run-time edit flags

Flag	Function
/HE	Halt on error. Testing halts when an error occurs.
/PE	Prompt on error. If an error occurs the test will not proceed until keyboard input is received.
/N	Specifies number of times to run the test. Defaults to “/N 1”, not needed with Testview.
/Q	No messages to console. Does not affect the other redirects.
/RR file	Redirect error, advisory, and status messages to file.
/RE file	Redirect error messages to file.
/RA file	Redirect advisory messages to file.
/RS file	Redirect status messages to file.
/SS <file>	Summarize loops and fails after each test. Optionally redirected to file or printer.
/SF file	Summary file redirection. Overrides default.
/W nn	Wait only <i>nn</i> seconds when reading keyboard. Waits indefinitely if the parameter is not present. May be any positive integer value. This is useful for running interactive tests unattended.

Run all tests

This option runs all enabled tests in the TEST menu. A submenu provides looping options.

Typing Ctrl-C or Ctrl-Break after starting the test aborts the sequence.

Loop count submenu

The Loop count submenu specifies the number of complete passes through the list of subtests. Choosing 0 or UNLIMITED means run indefinitely. The Test duration option will halt testing if set to a positive number.

Use the space bar to cycle through the following choices for this option:

- UNLIMITED
- 1 pass
- Blank (edit mode)

In edit mode, enter a positive integer (four-digit maximum).

Pressing <Enter> anywhere in the menu will start test execution with all options selected.

Test duration submenu

Test duration specifies the number of hours for testing. Actual testing time is also specified by the Loop count. Testing will stop if the loop count expires even if the specified duration has not elapsed.

Use the space bar to cycle through the following choices for this option:

- Unlimited
- 1 hour
- Blank (edit mode)

In edit mode, enter a positive number (four-digit maximum).

Pressing <Enter> anywhere in the menu will start test execution with all the options selected.

Subtest order submenu

The subtest order is the order in which the test will be run.

There are two subtest order choices:

- SEQUENTIAL runs tests in the order they are found in the menus.
- SHUFFLED runs each pass in a different random order.

Use the Space bar to alternate between the two choices.

Press <Enter> to start the tests with the options selected.

Use run-list

This option allows you to specify any sequence of tests and nested loops.

To specify modules, list the modules, separated by commas, within brackets as follows:

```
{1, 2, 3, 4, 5, 6}
```

Any number in the list can be replaced with a range, such as:

```
{1-5, 6, 9}
```

Looping can be indicated using the * symbol:

```
{1-5, 6*3, 9}*
```

An asterisk without a number following indicates an infinite loop.

The following example shows how to indicate lists within lists:

```
{1-5, {6.2, 6.3}*3, 9}
```

Spaces are not allowed in the run-list string. You can use text or numbers in the lists.

Run DOS command

The QUIT menu provides the means to exit Testview.

Invoking Testview

You can select the following options when invoking Testview without a specified PKG file: self-configuring, selection for PKG file list, or supply a PKG filename.

Testview command line flags

The Testview command line may use either Testview flags or run-time flags. **Table B-19** contains descriptions of the Testview flags.

Table B-19
Testview flags

Flag	Function
/AUTO [<filename>]	Auto integrate. This parameter causes Testview to perform a self configuration, display the hardware detected in the system, append the selfsens.cfg file to the PKG with a product code heading and invoke tests found in the PACKAGE.INTEGRATION_LIST parameter. If <filename> is present, the contents of the file specified are appended to the selfsens.cfg file.
/EXIT	Exit Testview when done. Runs all tests from the run-list, when used with /RUNALL, then exits without operator intervention.
/HOURS <nn.n>	Set the duration of testing. After <nn.n> hours has elapsed, testing will halt. Used in combination with other flags, such as /RUNALL or /SHUFFLE, <nn.n> becomes an upper limit, not a guaranteed duration.
/RUNALL [<nn>]	Run all enabled tests automatically. After tests are run the menus are up and ready for input. The <nn> argument indicates the number passes to be taken through the run list. Zero (0) or omitting <nn> indicates run forever.
/PC <product-code>	Product code selection. This flag will cause Testview to search the PKG file for a section labeled <product-code>. This section is then processed. All processed parameter definitions and test exclusions will over-ride previous settings from the PKG file.
/PKG <filename>	Causes Testview to use <filename> instead of t.pkg .

Table B-19
Testview flags (Continued)

Flag	Function
/SELF [<filename>]	Self sense. Probes the system hardware and sets the CFG parameters appropriately. The “sensed” hardware is presented to the operator for approval. The optional <filename> specifies a default PKG file when the BIOS ID matching fails.
/SHUFFLE [<nn>]	Shuffle test order. This flag causes all enabled subtests to be shuffled and run before the menus begin accepting input. The optional <nn> argument determines the number of times through the list. Omitting <nn> will cause the subtests to run forever, assuming no RUNALL flag.
/TITLE <i>text</i>	Over-rides the default banner at the top of the Testview screen.
End	

TESTVIEW.INI file

This file stores custom colors for specific menu groups in **Testview**.

The parameters determine the color of the menus and should not be modified with a text editor. The CONFIGURATION menu *Changing menu colors* option allows you to change colors without accessing this file.

Using the T Command

T (**t.exe**) is used to run tests from the DOS command line or batch file. The syntax for T is

```
t [<tests>] [<flags>]
```

The <tests> parameter can be a single test, module, or sequence of tests. If this parameter is omitted, it indicates that all tests be run.

Note: The <tests> parameter must always precede the <flags> parameter on the command.

The following table lists samples of T commands.

Table B-20
T commands

Command line	Function
T	Run all enabled subtests in all modules in the PKG files.
T FLOPPY	Run all enabled subtests in the floppy module.
T 1.1	Run the first subtest of the first module in the PKG file, if enabled.
T FLOPPY.1	Run the first subtest of the FLOPPY module, if enabled.
T 1-5	Run all enabled subtests of the first 5 modules in the PKG file.
T 1.3-1.7	Run all enabled subtests in the range 3 through 7 of the first module in the PKG file.
T {1,5,7}	Run all enabled subtests in modules 1, 5, and 7.

The *<flags>* parameter can be any number of valid flags which are either handled by T or passed on to the test.

The *<tests>* parameter has a rich syntax to allow for flexibility. The terse definition is as follows:

Note: The [...] and <...> are metasympols. Spaces are not allowed.

```
<tests> = <runlist> | <range> | <logical> [*<count>]
<runlist> = {<tests> [, <tests>]}
<range> = <logical> [-<logical>]
<logical> = <label> [.<label>]
<label> = string | number
```

When the hyphen (-) is used, enabled tests listed in the PKG file between (and including) the two specified tests are run.

When specifying a single subtest, it is necessary to separate the module name and the subtest name with a dot (.).

Both the module name and the subtest name may be specified as either a number or a string. Both of these mappings are determined by the **t.pkg** file. An empty runlist may be specified as {}.

The *<flags>* parameter is any subset of the following lists (see [Table B-21](#)). Control flags are passed to the test and handled by its library. Subtest flags are also passed to the test, but handled directly and may differ from test to test.

Table B-21
Flags parameter groups

Flag Group	Flags
(control)	/HE /PE /RR /RE /RA /RS /Q /SS /SF /
(subtest)	/P /V /L /CD ...

The flags in [Table B-22](#) are handled directly by T and not passed to the tests. In most cases the argument is optional

Table B-22
Flag description

Flag	Description
/?	Help. Prints out the "T.HLP" file
/AUTO [filename]	Auto integrate. This parameter causes T to perform a self configuration, display the hardware detected in system, add the selfsens.cfg file to the PKG with a product code heading and invoke any tests in the PACKAGE.INTEGRATION_LIST parameter in the PKG file.
/DT [nn]	Display tests. Lists the names and numbers of all subtests in the PKG file with a "+" or "-" for each subtest indicating if it is enabled. The default number of lines printed before pause is 20. The optional argument will override the default.
/GROUP chars	Restrict run-list. Selects only those subtests which have a character from the string <chars> in the attribute field. The predefined attributes such as "+" or "I" may be used. Or a group may be created by adding a single digit (such as "1") to the attribute fields of subtests to be included. CAUTION: this flag overrides the disable attribute. All tests in the group will run.
/N nn	Number of loops. This flag indicates the number of complete passes to be taken through the run-list. Zero(0) indicates run forever.
/OVER	Override disable flag. This flag will cause T to ignore the disable flag for any subtest called out specifically in the run-list. Subtests called out by a range, group or module in the run-list will not be overridden.

Table B-22
Flag description (Continued)

Flag	Description
/PC product-code	Product code selection. This flag will cause T to search the PKG file for a section labeled product-code. This section is then processed and any parameter definitions or test exclusions will override the previously processed contents of the PKG file.
/PKG filename	PKG file. This flag will cause T to process filename instead of "T.PKG."
/SELF [filename]	Self configure. This flag causes T to run all subtest-0's in the PKG file. These subtests create two files, "selfsens.cfg" and "display.cfg". The PKG file for a self sense is chosen by the BIOS ID. If the BIOS ID cannot be matched and no filename is specified, an error occurs. If filename is present, this is taken as the default PKG file. To over-ride the BIOS ID matching, use "/SELF /PKG filename".
/SHUFFLE [nn]	Shuffle the run-list. This flag causes the run-list to be shuffled on each pass. The optional nn argument determines the number of passes, unless the /N flag is used.
End	

Standard error codes

The error codes listed in **Table B-23** are generated by t.

Table B-23
T error codes

Error Code	Explanation
01300000	Error opening redirect file
01300001	Error opening PKG file
01300002	Error processing PKG or CFG file
01300003	Product code not found in PKG file
01300004	Test module EXE file not found
01300005	Error returned by spawn (while loading test)
01300006	Invalid return code from test
01300007	Memory error after running test
01300008	Error deleting file from RAM drive
01300009	Error renaming file on RAM drive
0130000A	Critical error handler return codes
0130000B	Parity interrupt occurred
0130000C	Invalid module name in command line run-list
0130000D	Invalid test name in command line run-list
0130000E	Second module name in run-list not valid
0130000F	Second test name in run-list not valid
01300010	Error checking free space on RAM drive
01300012	BIOS ID not found in any PKG file
01300013	Unexpected return code from subtest 0
01300014	Unable to open PKG file during auto-integrate
01300015	Unable to open selfsens.cfg file